"EXPRESS MAIL" Mailing Label No. ER 272264444 US Date of Deposit: November 13, 2003

AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING (AAA) SERVER

Technical Field

[0001] The present invention relates to communication systems. More particularly, and not by way of limitation, the present invention is directed to an Authentication, Authorization, and Accounting (AAA) server arranged to generate a master session identity that is usable to route queries containing the master session identity through a packet data network to the AAA server.

Background Art

[0002] The Remote Access Dial-In User Service (RADIUS) is an AAA client-server protocol. RADIUS is the *de facto* industry standard for remote access AAA, as well as an Internet Engineering Task Force (IETF) standard. In general, RADIUS is a network process that performs authentication, authorization, and accounting actions when a user logs in on a Network Access Server (NAS) with a dial-up client, or logs out from the NAS. Typically, a RADIUS server is used by Internet Service Providers (ISPs) to perform AAA tasks. AAA tasks include verifying the identity of an entity (authentication), determining whether a requesting entity is allowed to access a resource (authorization), and collecting information on resource usage for the purpose of trend analysis, auditing, billing, or cost allocation (accounting). The RADIUS server may also be used when controlled dial-up access is needed in a particular organization. A technical specification of basic features that are supported by all RADIUS servers can be found in RFC 2865/RFC 2138 and RFC2866/RFC 2139, which are hereby incorporated by reference herein.

[0003] Existing AAA servers perform RADIUS proxy functions using static configuration tables stored in the AAA server. The static tables do not change at run-time. The AAA

server typically uses the realm or the Access Point Name (APN) included in an incoming message to determine whether or not to forward the message. The realm is a part of the Network Access Identifier (NAI) that is included in the User-name attribute. According to Third Generation Partnership Project (3GPP) specifications, the APN is included as the value of the Called-Station-ID attribute. The address of the Network Access Server (NAS) is included in the NAS-IP-Address attribute.

[0004] In addition to the classic AAA functions, AAA servers may also perform session management functions such as hosting master sessions. A master session is a session created when a user is successfully authenticated by the system. The master session is terminated when the user logs off. The master session is created in a home AAA server, which hosts the data of the user to which the master session is tied. Anonymous master sessions, that is, sessions that are not bound to a user, may be created in any AAA server.

[0005] In addition to the classic RADIUS proxy functionality, there are a number of known alternative methodologies to enable a client to find the AAA server that hosts a given user. In one such methodology, the location of the AAA server is stored in an external repository such as a directory service. The directory service provides a discovery service to the client. When the client desires to access an AAA server, the client queries the directory to discover the appropriate AAA server. Afterwards, the client sends its request to the AAA server.

[0006] In another methodology enabling a client to find the AAA server that hosts a given user, all of the AAA servers that host user data (the AAA infrastructure) locate the specific AAA server when a request is received. RADIUS proxy functionality is implemented in each AAA server, and a load-balancing device randomly sends the request to any AAA server. When the AAA server receives the request, the server determines whether it is the appropriate AAA server to process the request. If not, the AAA server queries the directory service and, depending on the answer, forwards the request to the appropriate AAA server.

[0007] In another methodology enabling a client to find the AAA server that hosts a given user, the responsibility for finding the appropriate AAA server is again handled by the AAA infrastructure. A routing node contains a global directory database with the per-user

locations of the AAA servers, together with a RADIUS stack and the capability of performing RADIUS proxy functions. The client sends its requests to the routing node, which looks up the appropriate AAA server in its database, and forwards the request.

[0008] The existing static routing methodologies have several shortcomings. As the number of users grows, the number of AAA servers must grow to host the users' data. When an AAA server plays the role of session manager, the existing static routing methods (based either on internal tables or external look-ups) do not support the location of the appropriate AAA instance handling the user requests for a specific session. When a client wants to access a given master session, the only parameter that is available to look it up is the master session identity (ID) because no user identity is provided. However, the master session ID is a dynamic value generated at run time, and has special features (i.e., random, unique, not reusable, and unpredictable). The routing data used by the static routing methods is based on pre-configured static tables or databases that are not updateable at run time. Thus, there is no way to determine where a master session is stored.

[0009] Similar drawbacks to those stated above may be experienced when using a Lightweight Directory Access Protocol (LDAP) interface to access a central, and likely external, repository from a first-queried AAA server that is not in charge of a given user, in order to determine the appropriate AAA server where the query should be redirected.

[0010] It would be advantageous to have an AAA server that overcomes the above-described shortcomings. The present invention provides such an AAA server.

Summary of the Invention

[0011] The present invention overcomes the shortcomings of the prior art by introducing structure into the master session ID. While the master session ID remains as a random value overall, additional information indicating the situation of the master session is included within the actual master session ID.

[0012] In one aspect, the present invention is directed to an Authentication, Authorization, and Accounting (AAA) server in a packet data network. The AAA server includes means for authenticating a user; means for authorizing a service for the user when

the user accesses the network; and means for generating a session identity that comprises a unique random value that is opaque, unpredictable, and not simultaneously re-usable. Additionally, the means for generating a session identity includes means for structuring the session identity to include an identifier of the AAA server that is usable to route queries containing the AAA server identifier to the AAA server.

[0013] In another aspect, the present invention is directed to a system in a packet data network for routing queries to an appropriate AAA server. The system includes means for assigning a realm identifier to each of a plurality of AAA servers; means for creating a master session in a given AAA server; and means within the given AAA server for generating a master session identity that includes a session reference and the realm identifier assigned to the given AAA server. The system also includes means within the network for routing queries based on the master session identity to the given AAA server.

[0014] In yet another aspect, the present invention is directed to a method of routing queries to an appropriate AAA server in a packet data network. The method includes the steps of assigning a realm identifier to each of a plurality of AAA servers; creating a master session in a given AAA server; and generating by the given AAA server, a master session identity that includes a session reference and the realm identifier assigned to the given AAA server. Queries containing the master session identity are then routed to the given AAA server.

[0015] In yet another aspect, the present invention is directed to a method of routing queries to an appropriate AAA server in a packet data network. The method includes the steps of randomly generating in each of a plurality of AAA servers, a realm identifier, preferably of a fixed-length, that uniquely identifies each generating AAA server; creating a master session in a given AAA server; and generating by the given AAA server, a master session identity that includes a session reference and the realm identifier that identifies the given AAA server. Queries containing the master session identity are then routed to the given AAA server.

Brief Description of the Drawings

[0016] FIG. 1 is a flow chart of a first embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created;

[0017] FIG. 2 is a flow chart of a second embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created;

[0018] FIG. 3 is a flow chart of a third embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created;

[0019] FIG. 4 is a flow chart of a fourth embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created;

[0020] FIG. 5 is a flow chart of a fifth embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created;

[0021] FIG. 6 is a simplified diagram illustrating a first network configuration in which a plurality of AAA servers are configured to perform both proxy functions and home functions; and

[0022] FIG. 7 is a simplified diagram illustrating a second network configuration in which a specialized AAA server performs only proxy functions and serves as the AAA infrastructure front-end.

Detailed Description of Embodiments

[0023] When performing as a central session manager, an AAA server may authorize and account for a user based on certain master session parameters. The key parameter for identifying the master session is the master session ID. The master session ID is used as an identity token in the service layer. The master session ID is received in the user's equipment, and may be stored, modified, deleted, and so on. In order to prevent malicious use of the master session ID, certain measures need to be taken. Because of its exposure

in the application level, the master session ID is made to be a one-time random value that is unique, not reusable, unpredictable, and opaque. "Unique" means that two simultaneous ongoing master sessions shall not have the same master session ID. "Not reusable" means that the number of possible values is large enough to avoid repeating master session IDs previous utilized. "Unpredictable" means that it is not possible to guess a valid ongoing master session ID based on any number of previous IDs. Finally, "opaque" means that it is not possible to extract information from the master session ID.

[0024] In the following description, the actual master session ID is referred to as the "session reference" (session_ref) and the entire token or structure provided by the AAA server (regardless of its structure or composition) is referred to as the master session ID. There are a number of embodiments of the present invention, including different measures to increase security.

[0025] FIG. 1 is a flow chart of a first embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created. In the first embodiment, a realm identifier is assigned to each AAA server at step 11. In step 12, a master session is created in a given AAA server. At step 13, the session reference is encoded. A number of coding techniques may be utilized, including for example, Base64, UUcode, or hexadecimal coding. At step 14, the realm ID is appended to the encoded session reference. At step 15, the given AAA server then uses the encoded session reference with the appended realm ID as the master session ID. Thus, the master session ID takes the form: encodedsession_ref."@".realm_id. As an example, a master session ID may thus take the form:

[Base64(session_ref)]."@".realm_id

where "." indicates concatenation. Taking, for example, 2AFD4590BB0B1C21 as the session reference and aaa01 as the realm, the resulting master session ID is:

2AFD4590BB0B1C21@aaa01

[0026] FIG. 2 is a flow chart of a second embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created. In the second embodiment, a realm ID, preferably of a fixed-length, is randomly generated at step 21, and at step 22, the realm ID is associated with a given AAA server. At step 23, a master session is created in a given AAA server. At step 24, the session reference is encoded. At step 25, the realm ID is then appended to the session reference, or alternatively, the realm ID is placed at the beginning of the session ID (i.e., the session reference is appended to the realm ID). At step 26, the given AAA server then uses the combined session reference with the realm ID as the master session ID. Thus, a master session ID may alternatively take the form of:

encodedsession_ref.realm_id or realm_id.encodedsession_ref

Using the same session reference as the first embodiment, and using 2D54GA as a randomly generated realm ID, the resulting master session ID is:

2AFD4590BB0B1C21**2D54GA** or **2D54GA**2AFD4590BB0B1C21

[0027] FIG. 3 is a flow chart of a third embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created. In the third embodiment, the master session ID is encrypted in order to conceal the semantics of the master session ID. At step 31, a fixed-length realm ID is randomly generated, and at step 32, the realm ID is associated with a given AAA server. At step 33, a master session is created in a given AAA server. At step 34, the realm ID is then appended to the session reference, or alternatively, the realm ID is placed at the beginning of the session ID (i.e., the session reference is appended to the realm ID). At step 35, the combined session reference and realm ID is encrypted. Since the creator of the master session ID is the same entity as the one that looks up the specific master session, symmetric encryption may be utilized. All of the AAA servers share the same symmetric key. At step 36, the encrypted combined session reference and realm ID is

encoded. At step 37, the given AAA server then uses the encrypted combined session reference with the realm ID as the master session ID. Thus, the master session ID takes the form: encoded(encrypt(session_ref.realm_id)). As an example, a master session ID may thus take the form:

Base64(encrypt(session_ref.realm_id))

[0028] An advantage of this embodiment is that it prevents Denial of Service (DoS) attacks against specific AAA servers because whenever an AAA server receives a request, the server checks to ensure that the master session ID is not fake before forwarding the request. This advantage is gained at the expense of requiring more processing power in the AAA server to perform additional encryption and decryption operations required to create the identifier and look up the master session.

[0029] FIG. 4 is a flow chart of a fourth embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created. In the fourth embodiment, rather than encryption, a Keyed-Hasing Message Authentication Code (HMAC) is used at both ends of the communication to guarantee that a received master session ID is legitimate and to prevent DoS attacks. HMAC is a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any cryptographic hash function such as MD5, SHA-1, etc. in combination with a secret shared key. HMAC verification is much faster than encryption/decryption operations and, although based in a secret shared by both parties, it may re-use the common mechanism used by RADIUS servers and clients since both RADIUS servers and clients need to own a shared secret. Use of HMAC verification prevents attacks based on guessing the master session ID.

[0030] At step 41, a fixed-length realm ID is randomly generated, and at step 42, the realm ID is associated with a given AAA server. At step 43, a master session is created in a given AAA server. At step 44, the realm ID and the HMAC are then appended to the session reference. At step 45, the combined session reference, realm ID, and HMAC is encoded. At step 46, the given AAA server then uses the encoded combined session

reference, realm ID, and HMAC as the master session ID. Thus, the master session ID takes the form: encoded(session_ref.realm_id.HMAC)). As an example, a master session ID may thus take the form:

Base64(session_ref.realm_id.HMAC)

[0031] FIG. 5 is a flow chart of a fifth embodiment of a method of structuring a master session ID to include an identity of a given AAA server, when a corresponding master session is created. In the fifth embodiment, both encryption and HMAC are utilized. Encryption provides hiding of the structure while the HMAC is useful as a counter-tampering measure. At step 51, a fixed-length realm ID is randomly generated, and at step 52, the realm ID is associated with a given AAA server. At step 53, a master session is created in a given AAA server. At step 54, the realm ID and the HMAC are then appended to the session reference. At step 55, the combined session reference, realm ID, and HMAC is encrypted. At step 56, the encrypted combined session reference, realm ID, and HMAC is encoded. At step 57, the given AAA server then uses the encrypted and encoded combined session reference, realm ID, and HMAC as the master session ID. Thus, the master session ID takes the form: encoded(encrypt(session_ref.realm_id.HMAC)). As an example, a master session ID may thus take the form:

Base64(encrypt(session_ref.realm_id.HMAC))

[0032] FIG. 6 is a simplified diagram illustrating a first network configuration in which each of the AAA servers 61-64 is configured to perform both proxy functions and home functions. Each AAA server is configured with the routing tables needed to route a session-ID-based request to the appropriate AAA server instance. Thus, when a RADIUS request from a client 65 is randomly routed by a load balancer 66 to an AAA server such as AAA server 62, the AAA server 62 determines that AAA server 63 is the appropriate AAA server, and routes the request to AAA server 63.

[0033] FIG. 7 is a simplified diagram illustrating a second network configuration in which a specialized AAA server 68 performs only proxy functions and serves as the AAA infrastructure front-end. Only the specialized AAA server is configured with the routing tables needed to route a session-ID-based request to the appropriate AAA server instance. Thus, when a request from the client 65 is received by the specialized AAA server 68, the specialized AAA server determines that AAA server 63 is the appropriate AAA server, and routes the request to AAA server 63.

[0034] Additional protection measures may also be utilized with the present invention to ensure correct operation in the face of malicious attacks. As noted above in the third embodiment, encryption of the master session ID may prevent attacks based on knowledge of the structure of the master session ID. However, this embodiment imposes a requirement that the encryption/decryption key be distributed to all entities involved in the encryption and decryption processes. To reduce the impact of this requirement, the master session ID may be encrypted only when it is released to the user (i.e., between the browser and the border gateway). The master session ID is then carried on in plain text between the border gateway and the AAA server. Alternatively, distribution of the encryption/ decryption key can be avoided by having the AAA server decrypt the master session ID using an Application Programming Instruction (API).

[0035] Another possible type of attack is based on eavesdropping and sending a reply once the master session ID is determined. Eavesdropping/reply attacks cannot be prevented by encryption. Only partial solutions can be used. For example, the master session ID can be tied to the IP address of the user equipment, as long as this solution is used in a core network scenario in which the IP address of the user is trusted and available in the AAA server. For instance, a Cyclical Redundancy Check (CRC) value or digest (using a hash function) may be included in the master session ID structure. The AAA server adds that footprint to the master session ID, while the border gateway checks the IP address received in the request from the client. Alternatively, the IP address may be used to compute the HMAC of the master session ID.

[0036] As will be recognized by those skilled in the art, the innovative concepts described in the present application can be modified and varied over a wide range of

applications. Accordingly, the scope of patented subject matter should not be limited to any of the specific exemplary teachings discussed above, but is instead defined by the following claims.